Searching For Repeated Video Sequences

Tolga Can
Bilkent University
Computer Engineering
Ankara, TURKEY
tolgacan@cs.bilkent.edu.tr

Pinar Duygulu
Bilkent University
Computer Engineering
Ankara, TURKEY
duygulu@cs.bilkent.edu.tr

ABSTRACT

In this paper, we propose a new method to search different instances of a video sequence inside a long video and/or video collection. The proposed method is robust to view point and illumination changes which may occur since the sequences are captured in different times with different cameras, and to the differences in the order and the number of frames in the sequences which may occur due to editing. The algorithm does not require any query to be given for searching, and finds all repeating video sequences inside a long video in a fully automatic way. First, the frames in a video are ranked according to their similarity on the distribution of salient points and colour values. Then, a tree based approach is used to seek for the repetitions of a video sequence if there is any. Results are provided on a full length feature movie, Run Lola Run and on commercials of TRECVID 2004 news video corpus.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Search Process, Information Filtering

General Terms: Algorithms.

Keywords: Copy Detection, Story and Media Tracking, Bag-Of-Features, Tree Approach, Keyframe Extraction.

1. INTRODUCTION

While there is a growing amount of digital videos available in many sources, the current research on video retrieval does not go beyond image retrieval and discards the temporal information which makes videos distinct from images.

In searching for videos, most of the systems either use textual information provided in the form of manual annotations or speech transcript text; visual information extracted from video frames; or simple combination of both [21]. In all cases, the results are provided in the form of a single shot or a collection of shots. However, video shots are not independent and the valuable information is available with a sequence of shots rather than with individual shots.

MIR'07, September 28–29, 2007, Augsburg, Bavaria, Germany.

We argue that, for a video retrieval system to be distinct from an image retrieval system, it is important to search for video sequences rather than to search for individual shots. The Query by Example (QBE) approach from image retrieval can be adapted to find the similar sequences to a query sequence. Detection of similar sequences is important for retrieval purposes since it helps better indexing and summarization, and also important to reduce the huge amount of data since it eliminates the repetitions.

However, there are important issues to be considered: (i) the signal distortions due to digitisation or encoding, and different frame rates, (ii) variations in the order and the number of frames due to the editing of the sequences, (iii) dissimilarity of video frames due to view point and lighting changes. Based on these issues, we divide the application domains which require detection of identical or similar video sequences into three groups.

Copy detection: Growing volumes of broadcasted videos shared among different media resulted in a new requirement: detection and tracing of copies or duplicates. Detecting copies of videos is very important for copyright issues but difficult when the amount of data is large, resulting in a new challenge for Content Based Copy Detection [1]. The assumption in copy detection is that the videos are distorted due to digitisation or encoding, but not edited [12, 8].

Media tracking: Tracking a piece of media which is used in different times and in different sources is important. For example, companies want to monitor TV commercials to ensure that the commercials are broadcasted properly and to track competitor's commercials for planning their marketing strategies [7, 4]. Another example is the tracking of news stories in a single channel. It is common for the news channels to re-use the material as the related story develops by slightly editing the videos to remove or add material [5, 2]. In both cases, the repeated video sequences may have slight variations due to editing.

Story tracking: It is common for the important events to be captured with different cameras. In this case, although the event or more generally the story is same, the footage may be different since the camera positions and parameters may differ. Also, the footage may be edited differently to represent different perspectives. Similarly, in some movies, such as "Run Lola Run" and "Groundhog Day" some portion of the story repeats several times with different footage. In these cases, both the lighting conditions and view point of the camera may change resulting in large variations in the video sequences corresponding to the same instance.

Figure 1: Repeating sequences with large differences. These two sequences are so different since they are taken with different viewpoints. Our algorithm can detect this sequence since our features are viewpoint independent.

While a repeated video sequence is captured with the same single source in copy detection and media tracking problems, resulting in almost identical duplicates, in story tracking there are multiple sources causing largely varied sequences.

For example, although the two sequences in Figure 1 corresponds to the same story, the frames inside the stories are very different but since SIFT descriptors are robust to view point and illumination change this kind of sequences can be detected. (another example can be seen in Figure 11). Therefore, finding the similar frames for story tracking is challenging, and cannot be solved by using global features which are heavily experimented in finding similar video frames for copy detection and media tracking.

Recently, salient point based approaches are shown to be successful for matching objects and scenes in images with varying view points lighting conditions [16]. Going further, Bag-of-Features (BoF) approach, which is adapted from Information Retrieval literature, is applied for retrieval purposes based on the distribution of salient points [20].

In this study, we tackle media and story tracking problem, and provide a solution to detect similar video sequences with variations both due to editing, and also due to lighting and view point changes. We propose a tree based approach to find all instances of the repeating video sequences without requiring any manual effort to initialise a query. For handling the differences in frames due to view points and illumination changes we make use of salient points together with colour information. As used in Video Google [20], which finds the repeating objects in a video, to find sequences, we utilise the Bag-Of-Features approach to find the repeating video frames. The experiments are carried out on the movie "Run Lola Run" and on commercials of TRECVID 2004 data.

In the following first we discuss the related studies. Then we present the tree based approach for finding repeated sequences assuming that the rankings of the similar frames are given for each frame in the sequence. Then we discuss how these rankings can be found. We then present our experimental results.

2. RELATED WORK

Requirement to copy detection, media and story tracking is increasing parallel to growing amount of digital videos. There are several studies in these areas.

Most of the studies on copy detection focus mostly on signal distortions. They do not cope well with display formats. Kim et. al [12] proposed an algorithm to detect copies of a video clip based on sequence matching. They used both spatial and temporal similarities between sequences. Spatial similarity is based on 2*2 grid intensity averages. Distance among sequences are calculated by using intensity averages and temporal signatures of sequences.

Similarity measure calculation can affect copy detection results very deeply. Arun et. al [8], compares several image distance measures, Histogram Intersection, Hausdroff Distance, Local Edge Descriptors and Invariant Moments in their experiments. Their dataset contains exact copies and they propose that local edge descriptors followed by the partial Hausdorff Distance gives best result. Julien et. al [11] used a voting function based on use of the signal description, the contextual information and the combination of relevant labels. Instead of using SIFT descriptors, they propose a new descriptors based on 4 different spatial positions around interest points in 5 directions. 20-dimensional feature vector is extracted for each keypoint that are extracted by Harris detector. Their approach is more logical than using global features to detect sequences. Keypoints can give more accurate results to describe an image compared to global features.

Media tracking is the problem of keeping track of particular video usage. Arun et. al [7], propose a method for media tracking. They create an index table by using keyframes colours and gradients. To search a media, they first extract keyframes of segments in videos, encode that keyframe by using features and find similar ones by using previously created index table. Duygulu et. al. [2] track news events by finding the duplicate video sequences and identifying the matching logos. They use both visual cues of keyframes and textual information of shot transcriptions.

Sivic et. al. in [20], propose a method to object and scene retrieval, which finds all occurrences of a user outlined object in a video. They use affine co-variant regions and SIFT descriptors to identify objects. A visual vocabulary is created using K-means. Term Frequency Inverse Term Frequency (tfidf) vector is calculated for user outlined objects by using visual vocabulary. At retrieval stage, frames are ranked by using normalised scalar product of tfidf vectors.

Results of visual vocabulary based methods can change easily by using different approaches to create visual vocabulary. These effects are examined deeply in [19]. They compare Bag-of-Features approach for classification. Classification result can change depending on sampling strategy for keypoint detectors, visual vocabulary size and method used to define images based on visual vocabularies. Although, they report that random sampling gives better results for their classification results, we use affine co-variant regions since they are more useful and effective for our case.

Visual vocabulary technique is very effective but it does not use colour information and spatial layout of features. Lazebnik et. al. [14] propose a method to recognise scene categories based on global geometric correspondence. They repeatedly subdivide the image and compute histograms of local features. This method is not robust against geometric changes since it compares histograms of by one-to-one correspondence and subdividing image avert features to be robust against geometric changes.

Gauch et. al. [3], uses repeated characteristics of commercials to detect and track commercials in videos. As a first step, they extract extract shot boundaries, fades, cuts and dissolves by using RGB colour histograms and some thresholds to find temporal video segmentation. After this step, they use a hash table based on colour moments for frames. They detect sequences by using this hash table and voting scheme. They apply a filtering based on number of frames, relative lengths of shots and mean colour moment of

each shot. By using video sequence classification, they can classify sequences as commercial or non-commercials.

Most of the copy detection algorithms use string matching techniques as in [6]. Guimares et. al. propose a method based on the fastest algorithm of exact string matching, the Boyer-Moore-Hoorspool (BMH). They allow some small differences between two correspondent frames by adding a threshold to BMH. Also they modify shifts after a mismatch by allowing smaller distance to move the query pattern to the next alignment verification. Their new algorithm is faster than Longest Common Sub-string method but they are using some thresholds to find similarities.

Naturel et. al. [18] propose a method based on signatures generated from DCT of frames and hashing. First of all, shots are extracted from videos. For each shot a signature is calculated based on frames in that shot using DCT coefficients of frames. A hash table is created based on these signatures and used to find repeated sequences. For a query signature, all candidate shots are found from hash table and similarity value is calculated between candidate shot and query shot. Then sequence is detected based on this distance and a threshold value.

In addition to techniques discussed above, our method needs to detect shot boundaries to represent videos as efficient as possible by using less number of frames. There are several approaches for shot boundary detection. Koumaras et. al. [13] propose a method based on discrete cosine transform. Liu et al. [15] propose a shot boundary detection method based on temporal statistics using eigenspace updating method. In this method, histogram of current frame is compared with eigenspace model learnt from previous frames. Shot boundary is detected when model is not fit to the current frame well. Jeong et. al. proposes a method based on frame differences and histogram differences in [10]. In their approach, each frame is divided into MxM grids and intensity difference of consecutive frames are calculated as a first step. If this difference value is between two values based on two thresholds then their histograms difference is calculated and shot boundaries are detected based on another threshold.

3. SEQUENCE DETECTION

One of the main drawbacks with most of the mentioned systems is that, a hard threshold is put in finding duplicates of the frames and then sequences are found assuming that they contain the same set of frames. However, there are two problems with such approaches: first, it is difficult to define a single general threshold applicable to different characteristics of large number of frames; second, some frames can be missed due to wrongly selected thresholds causing gaps in the sequences.

In this study, we propose a threshold free approach to rank similar keyframes. Then we combine our threshold free ranking method and sequence detection method in a single step using a tree representation to find repetitive sequences.

In the following section, we present the proposed tree based approach to detect candidate sequences and then a pruning strategy to find the final sequences. We assume that for each frame, the rankings for the similar frames are provided as described in Section 4.

3.1 Tree based approach

We propose a tree representation which codes both the similarity and order information. The proposed approach does not require any query sequence to be given, and finds all repeating sequences automatically. This is performed by building a separate tree for each frame i in the video to find the candidate repeating sequences for a sequence starting from frame i. In our tree based approach, each path from root to leaf is considered as a sequence candidate. If a frame does not belong to a repeating sequence, then no candidate sequences will be produced by the tree and the frame will be marked as a non-sequence frame. Otherwise, the candidate sequence will further be examined in the pruning step to check whether the sequence is also approved with the sequences produced by the neighbouring frames.

The main idea of our method is that the frames of a sequence are repeated with similar periods. That is, if i^{th} frame of a sequence is repeated with period T then, $(i+1)^{th}$ frame of the sequence should also repeat with the same period T. That is, if a sequence is represented by a list of frames as $S_1 = \{f_1, f_2, ..., f_n\}$, then the repetition of that sequence after T frames should be represented by a list of frames as $S_2 = \{f_{T+1}, f_{T+2}, ..., f_{T+n}\}$. (Here f_i corresponds to the i^{th} frame in the video). This means that if there are T frames between the first frame of sequence S_1 and first frame of sequence S_2, than a similar distance should appear for all the other frames of the sequences. We refer this constraint as the **distance constraint**.

However, due to missing or additional frames, and since the order of frames may slightly change from one sequence to another, the strict distance constraint is not satisfied in real situations. Instead we modify the constraint by adding a neighbourhood information. We assume that two similar frames could be the corresponding frames in the repeating sequences if they are placed with distances $T \pm \delta$ where δ is a small number. This δ ensures that sequence candidates are consistent in time. δ plays an important role to find sequences with edited or missing frames. We can find all repetitions if **distance constraint** is satisfied. For example, if frame positions for two consecutive frames f_i and f_{i+1} are i and $i+1$, then we allow editing new frames between f_i and f_{i+1} at most δ frames. This will result in new position of frame f_{i+1} as $f_{i+1+\delta}$. In the same way, we allow removal of δ frames. In the new situation, we use frame $f_{i+1+\delta}$ as the frame f_{i+1}. These kind of missing or editing conditions are allowed by our **distance constraint** for sequences discarding its length. Only drawback of δ value is that if δ is set too large, then number of sequence candidates will increase and eliminating sequence candidates will take longer time. We eliminate candidate sequence by sequence pruning described as Section 3.2.

The tree is constructed for frame i as follows. First, the frame i is placed as the root node (level 0). The nodes in the first level corresponds to the frames which are similar to frame i. Then, for each node in the first level, the corresponding children nodes are constructed in the second level, for the frames which are similar to the $(i+2)^{th}$ frame and appearing within a distance δ from their parents. In general, the $(n+1)^{th}$ level of the tree is constructed such that the nodes in that level corresponds to the frames which are similar to the $(i+n)^{th}$ frame in the sequence and placed with distance δ from the frames in their parent nodes. Each path from root to leaf node is considered as a sequence candidate.

In some level n, we may not be able to insert nodes to any paths since the distance constraint is not satisfied. However, that does not disallow adding new nodes in the next levels. This approach is important for dealing with missing frames. Here the choice of δ is important since large values corresponds to allowing large gaps which usually does not happen in sequences, and small numbers cannot deal with small number of missing frames. In the algorithm, this could be implemented as adding an empty node to the paths. In the experiments δ is chosen as 7.

This tree based approach has two problems. First of all, if the video has N frames then each frame will have N-1 similar images, listing all the similar frames causes a huge tree which is impossible to handle. Even the distance constraint is not sufficient to reduce the number of nodes, since for each node it will limit the number of children nodes with only 2δ. In Section 4 we discuss methods to limit the number of similar images differently for each frame.

The second problem is that, in the current form, there is no condition to stop the tree for growing and therefore for each path in the order of $N-i$ frames should be investigated to be added for the i^{th} frame in the video. However, note that as mentioned above, for some levels it is possible not to add any node to any paths since the similar frames do not satisfy the distance constraint. We use this fact and stop investigating the paths if for consecutive L levels it is not possible to add any new node to those paths. In the experiments we choose L as 10. This is a meaningful number since we do not expect the sequences to have gaps larger than ten frames.

The above steps are applied to each frame in the video, and for each frame the paths with lengths more than 3 are selected as candidate sequences. These candidate sequences are further pruned to see whether they are consistent with the candidate sequences found for the neighbouring frames.

The approach is simulated on an example given in Figure 2. Here, assuming that we have a sequence including the frames f0,f1,f2,f3 and f4, we would like to find the candidate repeating sequences. The figure shows the tree construction for frame f0. We assume that, it has a list of similar frames which are f50, f700, f600, f254 and f327 in the ranked order. In the first level, these similar frames to f0 is placed. Then, in the second level we insert the similar frames of f1 which follows f0 in the sequence. Similar frames of f1 are f51, f255, f1000 and f602 and f603. Obeying the distance constraint, these frames can only be added as the children to the nodes in the first level if they are in a δ neighbourhood. Here, we choose δ as 7. Therefore, only f51, f602, f603 and f255 can be placed in the second level of the tree as the frames which are similar to f1 and in the δ neighbourhood of f50, f600, and f254. Note that, f600 has two children f602 and f603. The third level is constructed similarly, by only placing f52 and f604. Note that in the third level, there is no similar image of f1 in the neighbourhood of f254, therefore this path is skipped in that level. The node for f256 is inserted to the path of f254 while considering the similar frames of f3 in the next level. In levels three,four and five the frames f604, f605 and f607 are inserted under both f602 and f603 causing two different paths corresponding to two different candidate sequences. The correct sequence is selected in the pruning part.

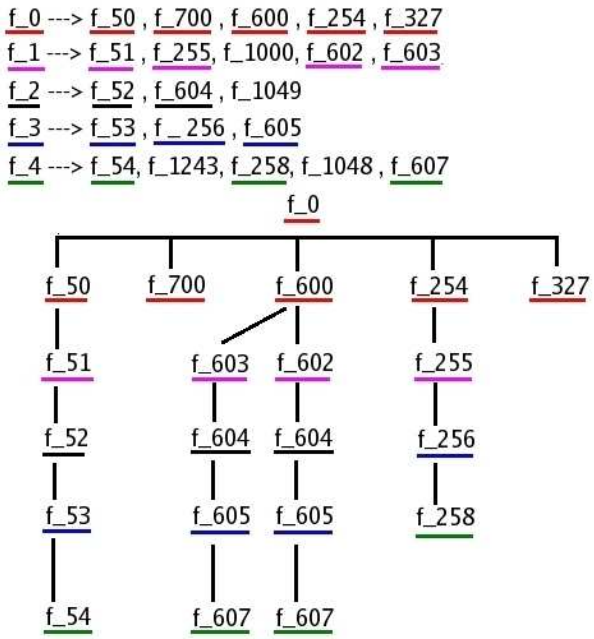

Figure 2: An example of tree construction for a frame f0. First five lines gives similarity rankings for consecutive five frames. The underlined colours in the tree, which are matched with the colours in the similarity rankings, shows by which frame it is added.

3.2 Pruning Sequences

We consider each path from root to leaf as a candidate sequence. However, in the set of candidate sequences, there are also many false alarms which are needed to be eliminated.

There are two types of false alarms. First type is the ones which are actually sub-sequences of a longer sequence. This type of false alarms occur since for a sequence length with length L, there are L-2 sub-sequences with starting and ending frames $[i,(i+L)],[(i+1),(i+L)],\ldots,[(i+L-2),(i+L)]$ if we let all sequences with length greater than 3 to be candidate sequences.

Second type of false alarms are the ones that are not actually sequences but decided as sequences. Since our definition of repeating sequences requires similarity of consecutive frames, because of the insufficiency of the feature representations, two sequences may be very similar when the visual features are considered but actually may not even be sequences by themselves

These two type of false alarms require different solutions. For the first type, we track sequence candidates for consecutive frames and try to find sequence's actual starting and ending positions. If candidate sequence is not repeated inside the other sequences found for the neighbouring frames, then that candidate sequence is labeled as a false alarm. Among the candidate sequences which repeats in the other sub-sequences the longest one with the farthest starting and ending points are taken as the final sequence, and the others are eliminated.

To eliminate the second type of false alarms, which are more commonly encountered, we apply a one-to-one match constraint. We require that two sequences S_i and S_j to be repeated sequences, both S_j should be found in the candidate list of S_i, and also S_i should be found in the candidate list of S_j. Note that, since the similarity rankings are different for each frame, in the case of false alarms it is unlikely to have the candidate sequences in both direction to be constructed. As expected, one-to-one constraint largely reduces the number of false alarms.

4. RANKING SIMILAR FRAMES

For the sequence detection algorithm to be successful, it is very important to capture the similarities between frames correctly. We use SIFT descriptors extracted from affine covariant regions to be robust to view point and illumination changes, but also incorporate the colour information in the form of HSV statistics extracted from fixed sized grids since colour is also a valuable information in most of the cases.

We detect two types of viewpoint co-variant regions for each frame as used in [20]. First one, called as Shape Adapted Region, is constructed by elliptical shape adaptation around an interest point. Second one, called as Maximally Stable Region, is constructed by selecting areas from an intensity watershed image segmentation.

We extract 128-dimensional vector SIFT descriptors [17] from each region. We also add the location of keypoints and transformation coefficients to SIFT descriptors, obtaining a 133-dimensional descriptor for each region.

In order to rank the images based on SIFT descriptors, we adapt the Bag-Of-Features approach as in Video Google [20]. In this approach, which is adapted from Information Retrieval, each image is described by a distribution of vector quantised form of the visual features, which are called as Bag-Of-features. In order to apply it to our problem, we vector quantised the 133 sized feature vectors into clusters using K-means. Since Maximally Stable regions and Shape Adapted regions corresponds to different characteristics, we vector quantise the features of them separately into 1000 and 500 clusters respectively. These values are found by our experiments. As a result of our experiments, we decided to choose $numberOfSIFTDescriptors/1000$ as cluster count for K-means. After quantisation, we construct a single vector of size 1500 for each frame, showing the distribution of clusters obtained from both. In order to fully utilise Information Retrieval techniques we weight the frequencies of the clusters using 'term frequency inverse document frequency (tf-idf)' which is computed as follows

$$tfidf = \frac{n_{id}}{n_d} \log \frac{N}{n_i} \quad (1)$$

where, n_{id} is number of occurrences of term i in document d, n_d is total number of terms in document d, N is the total number of documents in database and n_i is the number of documents in database containing term i.

Then, the similarity of frames based on SIFT descriptors of the salient regions are found by normalised scalar product of tfidf vectors by using the following equation. We refer this distance as D_1.

$$D_1(f_1, f_2) = \frac{tfidf(f_1) * tfidf(f_2)}{norm(f_1) * norm(f_2)} \quad (2)$$

where, $tfidf(f_1)$ is tf-idf vector of frame f_1 and $norm(f_1)$ is the norm of tfidf vector of frame f_1.

Although SIFT descriptors extracted from Maximally Stable and Shape Adapted regions are important for allowing view point and illumination changes, we noticed that they are not sufficient to correctly capture the similarities, and we also incorporate the colour information.

We use 5x7 grid HSV statistics to represent colour. Each frame is divided into 5x7 grids and mean and standard deviation of each band is calculated for each grid. We obtain 210-dimensional vector for each frame for colour data. Then Euclidean distance is used to define the similarity of two frames based on HSV statistics. We call this distance as D_2.

Neither SIFT descriptor based similarity values nor HSV statistics based similarity values are perfect. Also none of them has higher priority. For these reasons, we decided to combine these similarity values by equal weights and obtain a single distance value $D(f_i, f_j) = D_1(f_i, f_j) + D_2(f_i, f_j)$ for each frame pair (f_i, f_j). Note that the distances are normalised before combination.

As discussed previously, we do not want to put a single threshold to select the identical or similar images. However, using all the frames as similar frames is also not feasible. To handle this problems, for each frame f_i, first we rank all the images according to distance D, and then seek for a jump in the distances to separate the similar instances from the others. We eliminate the different frames by using peaks on similarity values. We apply a filter to sharpen peaks and find the maximum peak to ignore different frames. When the peak is at a number less than 10, we take 10 as the peak position. This approach allows us to reduce the number of similar frames without loosing the correct ones as seen in Figure 3.

Figure 3: Ranking results for a query image. First image is query image and others are most similar images. Titles of images show their dissimilarity values. In our approach we find peak position after 7^{th} image and take first 10 image as similar to query image.

5. EXPERIMENTAL RESULTS

Experiments are carried out on a full length movie, Run Lola Run, and commercials of TRECVID news corpus. In both datasets, repetitions are not exactly same but similar. There are some illumination changes and also scene changes. We use the keyframes provided by NIST for TRECVID corpus. We label keyframes of TRECVID as commercial or news. For the movie we extract the keyframes using the following approach. We consider the movie as a collection of keyframes only so that distance constraint is not affected by keyframe extraction.

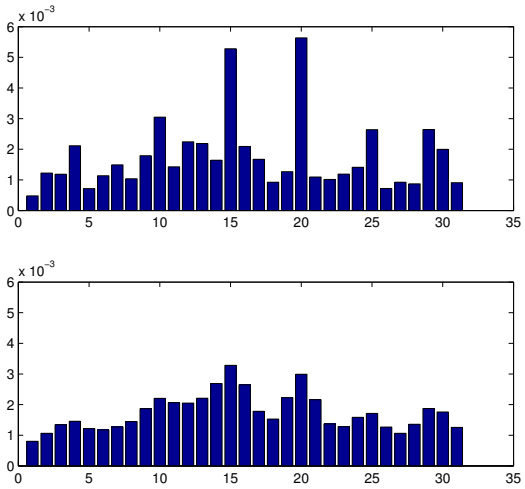

Figure 4: Canny Edge Histogram differences (above) for consecutive images and its filtered version(below).

5.1 Keyframe Selection

Each video contains almost 25-30fps in general so that a one hour long film contains more than 100000 frames at total. This number can be feasible for most of the applications but most of these keyframes are same or very similar. We can remove these similar frames without loosing any knowledge about general structure of the videos. Instead of using repetition of same frames several times, we can discard these repetitions and use the one as a representative frame, keyframe, among similar ones.

We need to find shot boundaries for keyframe extraction. There are several approaches for shot boundary detection given in [10], [9], [15]. We use a similar approach as used in [10] since none of the methods can give exact results for shot boundary detection.

First step in keyframe extraction is finding shot boundaries. First frame or last frame or median of all frames in a shot can be used as a keyframe for that shot. We use first keyframe of each shot as a keyframe.

We try to use a method independent of thresholds. Our shot boundary detection algorithm works based on colour and edge histogram differences. This is a two-pass algorithm.

In the first pass, we calculate colour and edge histogram differences for consecutive frames. We know that at shot boundaries these difference values must have higher values than surrounding differences, some kind of peaks or local maximas. We use both colour and edge histograms because

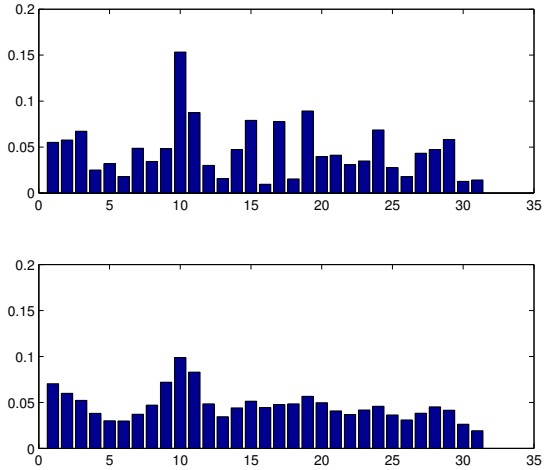

Figure 5: Colour Histograms differences (above) for consecutive images and its filtered version(below).

using only one of them can give lots of false alarms for shot boundaries. Common peak positions in difference values are taken as shot boundaries but there are too many peak locations in raw difference values. We need to remove some small peaks without using a threshold value.

We decided to apply a smoothing filter to difference values. For smoothing, we used an approximation of Gaussian Smoothing with mean 3 and σ 1. This helps us to remove some small differences. Results of this filtering can be seen at Figure 5 and Figure 4. We can find peak positions, where difference values start to decrease just after an increase, after these smoothing step. Common peak position in edge and colour histograms are considered as shot boundaries as a result of first pass.

In the second-pass, we extract keyframes according to shot boundaries. We consider first frame of each shot as a keyframe.

5.2 Feature Comparison

Three kind of features are used to compare results, SIFT descriptors, HSV statistics and combination of SIFT and HSV statistics. For comparison, we use precision and recall values for TRECVID dataset and precision values for Run Lola Run. Run Lola Run contains more longer sequences than TRECVID dataset.

Table 1: Sequence Detection precision values for Run Lola Run Movie.

Method	Correct Det.	False Det.	Precision
SIFT Descriptor	89	19	0.82
HSV Statistics	55	18	0.75
Combination	105	13	0.89

While SIFT descriptors are widely used for object matching solely they are not very sufficient in our case as in Figure 6. These kind of false alarms can be removed by using colour information of images. However, single use of HSV values also produce many false matches as in Figure 7. On the other hand simple combination of SIFT descriptors and HSV statistics gives the best performance for Run Lola

Run movie. The comparison of features are given in Table 1 and Table 2, for Run Lola Run and TRECVID, respectively. Since TRECVID dataset mainly contains commercials, HSV statistics gives better results because of colourful structure of commercials. But this is not a drawback for our tree based method because our approach mainly deals with ranking results instead of features. Any feature set according to dataset can be used for ranking.

Table 2: Sequence Detection Results for TRECVID dataset.

Method	Precision.	Recall.
SIFT Descriptor	0.92	0.71
HSV Statistics	0.98	0.76
Combination	0.91	0.74

Some example sequences using combination of two features are shown in Figure 8, Figure 9, Figure 12, Figure 10 and Figure 13 for Run Lola Run, and in Figure 14 and Figure 15 for commercials of TRECVID. As shown in the figures, the proposed method is able to capture both the differences in the frames due to view point and illumination changes and the differences in the number and order of frames in the sequences.

5.3 Complexity Analysis

We did not analyse preprocessing part of our algorithm since it is done only once. This part includes shot boundary and keyframe extraction, local interest points and descriptor extraction, visual term preparation and tfidf calculation. After preprocessing step, our tree-based approach creates a tree for each image in our dataset. Sequences are found according to created trees. If a frame is a member of a sequence length m and maximum number of similar frames for one frame is d then running time for creation of one tree is $O(m*d*log(d*m))$. If frame is not a member of a sequence, tree creation takes $O(10*d*log(10*d))$. In our approach, we create a tree for each frame so that running time of tree creation is $O(N*m*d*log(m*d))$, where N is the number of images in our database, d is the maximum number of similar images for one image and m is the length of longest sequence. Our method's space complexity is $O(m*d)$ since we need to store only one tree of a keyframe at a time.

However, note that while N is in the order of 6000, d is in the order of 50 and m is in the order of 15. For Run Lola Run which contains 5922 keyframes, the algorithm run on a P4 1 GH machine in 1148 seconds.

6. CONCLUSION

In this study, we propose a method to search for the similar instances of a sequence inside a long video. Unlike most of the current studies on video copy detection and media tracking, the proposed method is robust to view point and illumination changes which may occur since the sequences are captured in different times with different cameras, and to the differences in the order and the number of frames in the sequences which may appear due to editing. The algorithm does not require any query to be given for searching, and finds all repeating video sequences inside a long video or a video collection in a fully automatic way.

The experimental studies show that, the algorithm is successful in media tracking, specifically commercial tracking, and also in story tracking which is a more difficult task. As a future study, the approach will be tested on tracking of news stories in different channels.

7. REFERENCES

[1] CIVR. Video copy detection evaluation showcase.

[2] P. Duygulu, J.-Y. Pan, and D. A. Forsyth. Towards auto-documentary: Tracking the evolution of news stories. In *Proceedings of the ACM Multimedia Conference*, 2004.

[3] J. M. Gauch and A. Shivadas. Identification of new commercials using repeated video sequence detection. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 3(2):1252–1257, 2005.

[4] J. M. Gauch and A. Shivadas. Finding and identifying unknown commercials using repeated video sequence detection. *Comput. Vis. Image Underst.*, 103(1):80–88, 2006.

[5] J. M. J. Gauch and A. Shivadas. Visual story tracking in video news broadcasts, under review.

[6] S. J. F. Guimaraes, R. K. R. Coelho, and A. Torres. Counting of video clip repetitions using a modified bmh algorrithm : Preliminary results. *Multimedia and Expo, 2006 IEEE International Conference on*, pages 1065–1068, 2006.

[7] A. Hampapur and R. Bolle. Feature based indexing for media tracking. In *International Conference on Multimedia and Expo*, New York, NY, USA, 2000.

[8] A. Hampapur and R. Bolle. Comparison of distance measures for video copy detection. In *International Conference on Multimedia and Expo*, New York, NY, USA, 2001.

[9] G. Jaffre, P. Joly, and S. Haidar. The samova shot boundary detection for trecvid evaluation 2004.

[10] I.-S. Jeong and O.-J. Kwon. Video shot boundary detection using relative difference between frames. *Optical Engineering*, 42(3):604–605, 2003.

[11] V. G.-B. Julien Law-to, Olivier Buisson and N. Boujemaa. Robust voting algortihm based on labels of behavior for video copy detection. In *Proceedings of the ACM Multimedia*, 2006.

[12] C. Kim and B. Vasudev. Spatiotemporal sequence matching for efficient video copy detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):127–132, 2005.

[13] H. Koumaras, G. Gardikis, G. Xilouris, E. Pallis, and A. Kourtis. Shot boundary detection without threshold parameters. *Journal of Electronic Imaging*, 15, 2006.

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.

Figure 6: An example false sequence that is found by using SIFT Descriptors.

Figure 7: A false sequence found by using HSV statistics. First two rows represent real sequence. Second rows is found as repetition of real sequence.

Figure 8: An example sequence found by SIFT and HSV combination. Number of frames and frames in sequences are different.

Figure 9: An example sequence. Number of frames in sequences are same but frames are not exactly same.

Figure 10: An example of slightly different sequences.

Figure 11: An example sequence taken with different camera angles.

Figure 12: An example sequence that contains different keyframes because of camera position and keyframe extraction method.

Figure 13: Longest sequence found by our method.

Figure 14: An example sequence from TRECVID dataset. Second and third frames are different in real sequence and repetition.

Figure 15: An example sequence from TRECVID dataset. Number of keyframes are not same for two sequences.

[15] X. Liu and T. Chen. Shot boundary detection using temporal statistics modeling. *Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on*, 4(4):3389–3392, 2002.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[17] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.

[18] X. Naturel and P. Gros. A fast shot matching strategy for detecting duplicate sequences in a television stream. In *CVDB '05: Proceedings of the 2nd international workshop on Computer vision meets databases*, pages 21–27, New York, NY, USA, 2005. ACM Press.

[19] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision*. Springer, 2006.

[20] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1470, Washington, DC, USA, 2003. IEEE Computer Society.

[21] C. G. Snoek and M. Worring. Multimodal video indexing: A review of the state-of-the-art. Technical Report 2001-20, Intelligent Systems Lab Amsterdam Group, University of Amsterdam, December 2001.